Vivado从此开始（To Learn Vivado From Here）

本书围绕Vivado四大主题

- 设计流程
- 时序约束
- 时序分析
- Tcl脚本的使用

作者：高亚军（Xilinx战略应用高级工程师）

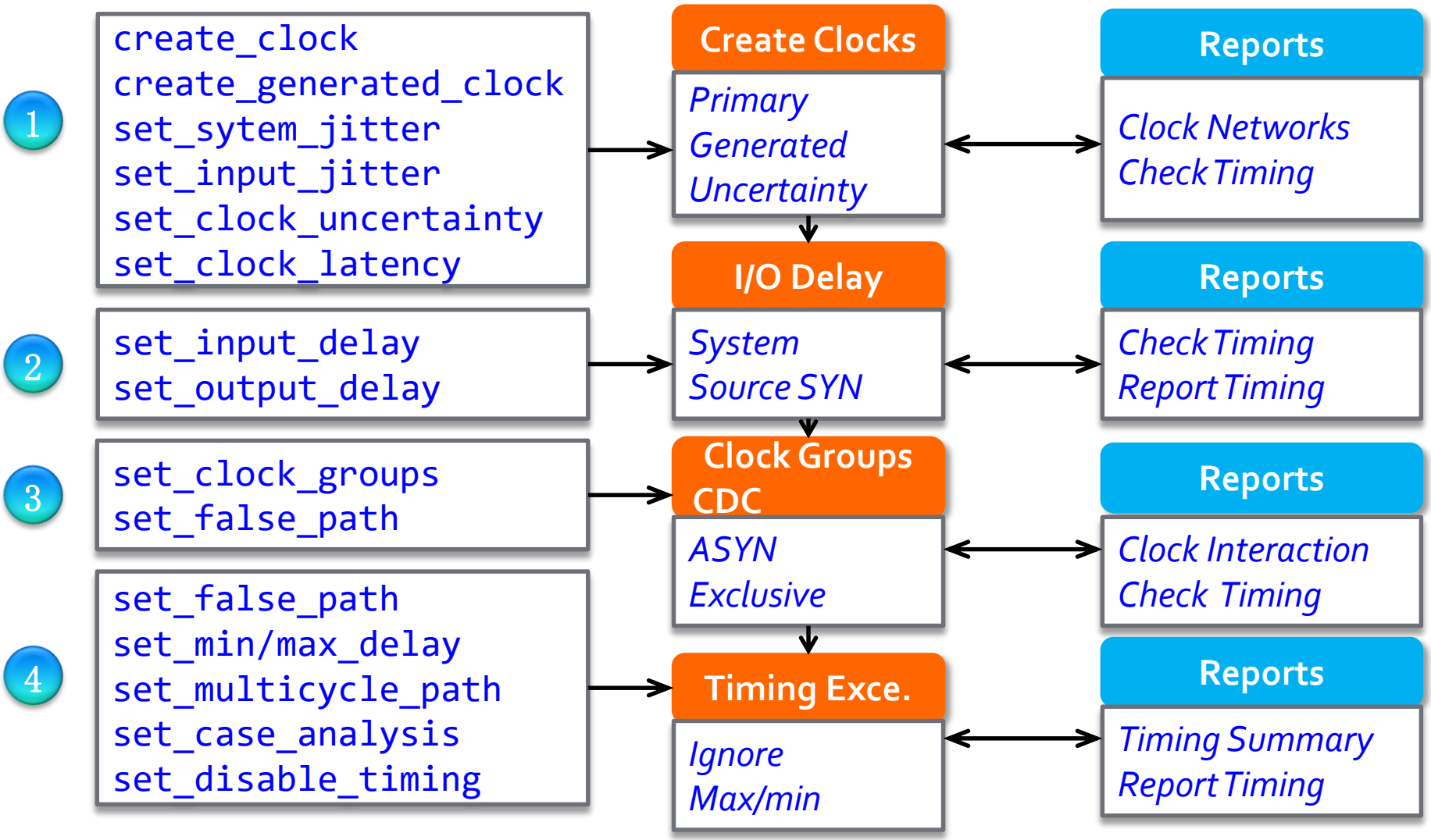
- 2012年2月，出版《基于FPGA的数字信号处理（第1版）》
- 2012年9月，发布网络视频课程《Vivado入门与提高》
- 2015年7月，出版《基于FPGA的数字信号处理（第2版）》
- 2016年7月，发布网络视频课程《跟Xilinx SAE学HLS》

- 内容翔实全面：涵盖Vivado所有基本功能
- 讲解深入浅出：结合大量案例，帮助读者加强对基本概念的理解
- 描述图文并茂：给出具体操作步骤，易于快速动手实践

UltraFast Design: Timing Constraint

Lauren Gao

Defining Timing Constraints in Four Steps

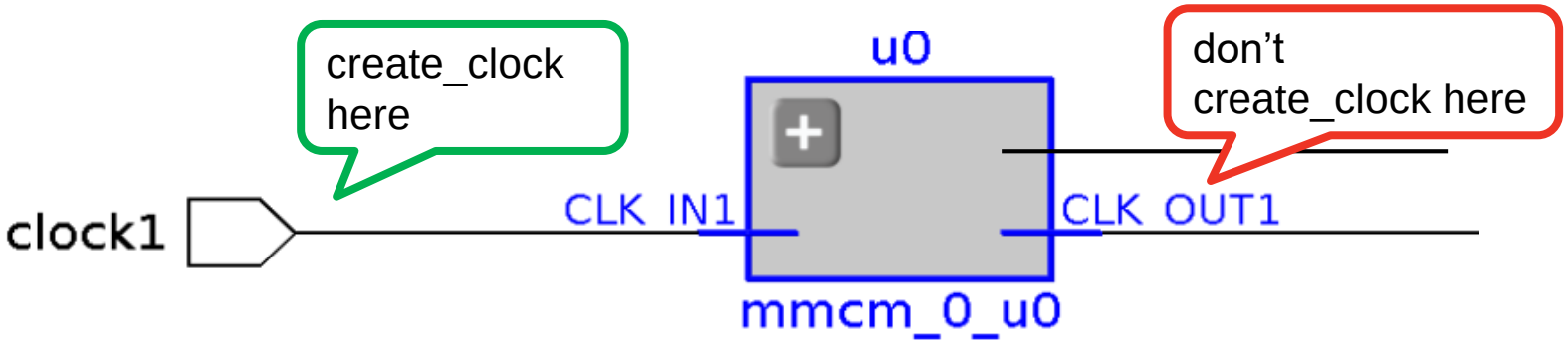


Method to Create Good Constraints

- **Create clocks and define clock interactions**
  - Four-step guideline
- **Set input and output delays**
  - Beware of creating incorrect HOLD violations
- **Set timing exceptions**
  - Less is more!
  - Beware of creating incorrect HOLD violations
- **Use report commands to validate each step**

Clock Ground Rules

- **For SDC-based timers, clocks only exist if you create them**
  - Use `create_clock` for primary clocks
- **Clocks propagate automatically through clocking modules**
  - MMCM and PLL output clocks are automatically generated
  - Gigabit transceivers are not supported. Create them manually.



- **Use `create_generated_clock` for internal clocks *(if needed)***
- **All inter-clock paths are evaluated by default**

Four Steps for Creating Clocks

- **Step 1**
  - Use **`create_clock`** for all primary clocks on top level ports
  - Run the design (synthesis) or open netlist design
- **Step 2**
  - Run **`report_clocks`**
  - Study the report to verify period, phase and propagation
  - Apply corrections to your constraints *(if needed)*

```
Attributes
  P: Propagated
  G: Generated

Clock            Period  Waveform        Attributes  Sources
sys_clk          10.000  {0.000 5.000}   P           {sys_clk}
pll0/clkfbout    10.000  {0.000 5.000}   P,G         {pll0/plle2_adv_inst/CLKFBOUT}
pll0/clkout0     2.500   {0.000 1.250}   P,G         {pll0/plle2_adv_inst/CLKOUT0}
pll0/clkout1     10.000  {0.000 5.000}   P,G         {pll0/plle2_adv_inst/CLKOUT1}
```

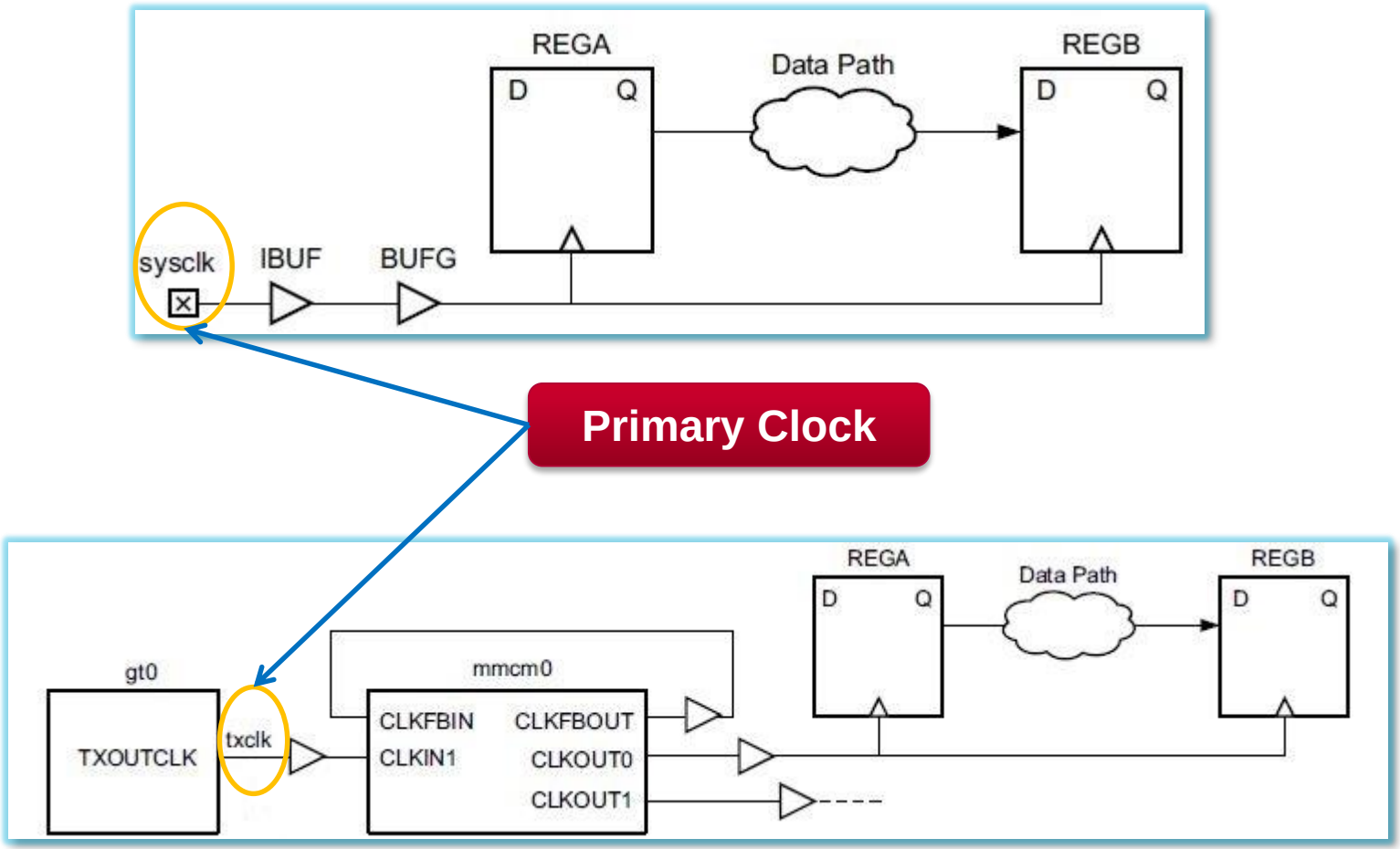
Four Steps for Creating Clocks *(continued)*

- **Step 3**
  - Evaluate the clock interaction using **`report_clock_interaction`**

    BEWARE: All inter-clock paths are constrained by default!
  - Mark inter-clock paths (Clock Domain Crossing) as asynchronous
    - Make sure you designed proper CDC synchronizers
    - Use **`set_clock_groups`** (preferred method to set_false_path)

      BEWARE: This overrides any set_max_delay constraints!
  - Do you have unconstrained objects?
    - Find out with **`check_timing`**
- **Step 4**
  - Run **`report_clock_networks`**
  - You want the design to have clean clock lines without logic
    - Tip: Use clock gating option in synthesis to remove LUTs on the clock line

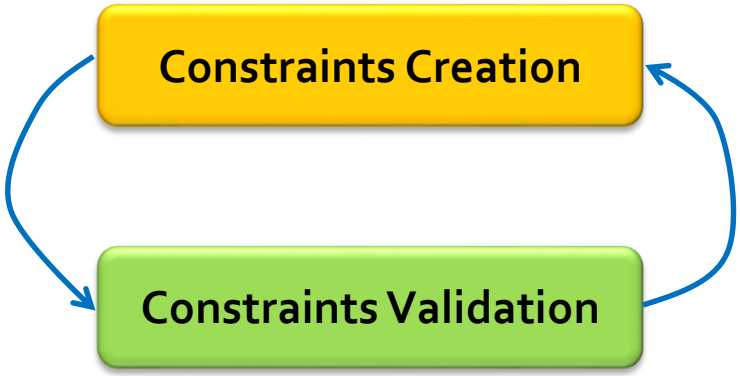
What is the Primary Clock



Adjusting Clock Characteristics

- **Jitter**
  - Input jitter: **`set_input_jitter`**
  - System jitter: **`set_system_jitter`**
- **Additional uncertainty**
  - **`set_clock_uncertainty`**
  - Add extra margin the timing paths of a clock or between two clocks
  - This is also **the best and safest way to over-constrain** a portion of a design without modifying the actual clock edges and the overall clocks relationships

Constraints Validation

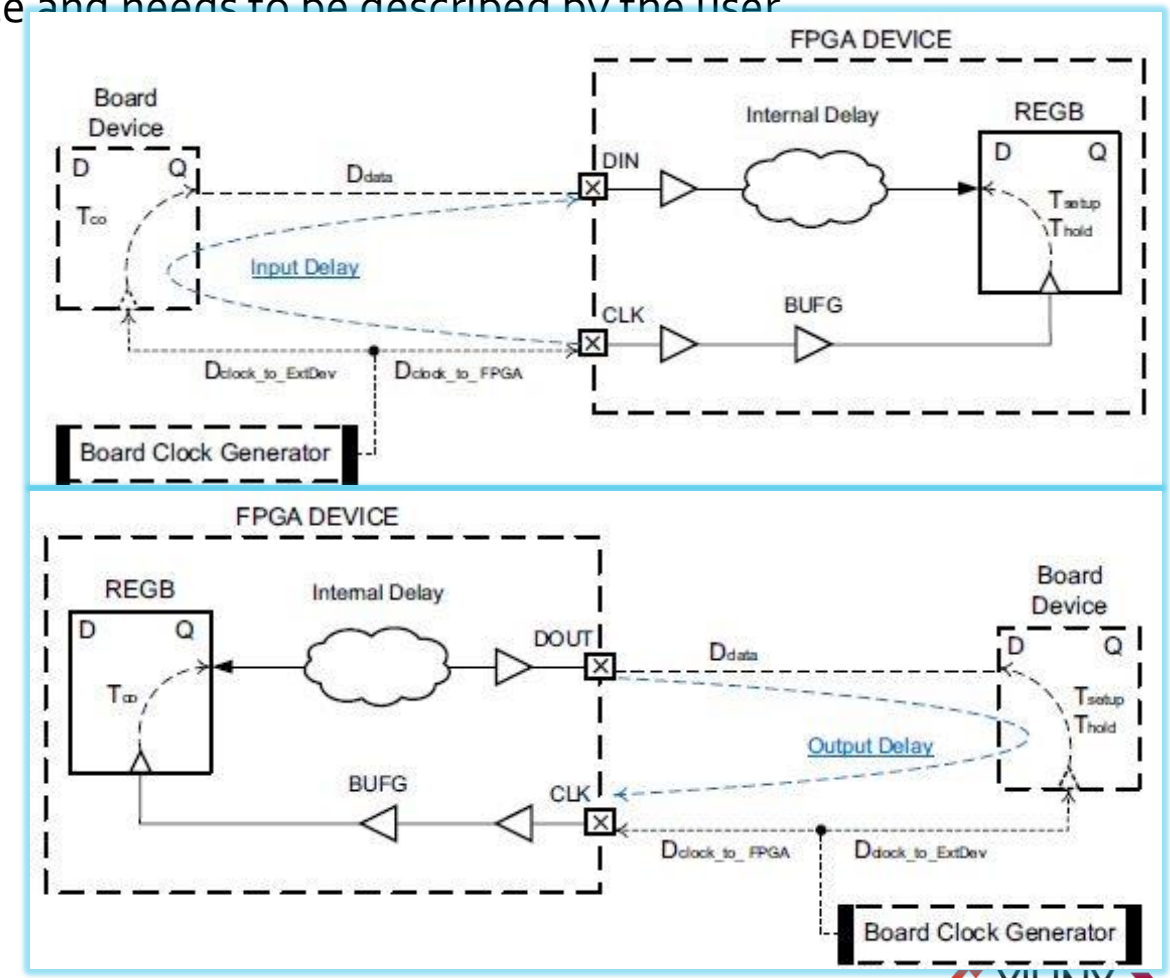



```
# Check if there are endpoints that are missing a constraint
check_timing
check_timing -override_defaults no_clock
# Determine the source of missing clocks
check_timing
report_clock_networks
# Validate clock characteristics
report_clocks
report_property [get_clocks wbClk]
```

Constraining Input and Output Ports

System Level Perspective

– The I/O paths are modeled like any other **reg-to-reg** paths by the Vivado Design Suite timing engine, except that part of the path is located outside the FPGA device and needs to be described by the user



Timing Exceptions Guidelines

- Use a limited number of timing exceptions and keep them simple whenever possible
  - The runtime of the compilation flow will significantly increase when many exceptions are used, especially when they are attached to a large number of netlist objects

NOT Recommended

```
set_false_path -from [get_ports din] -to [all_registers]
set_false_path -from [get_ports din]
set_false_path -from [get_ports din] -to [get_cells blockA/config_reg[*]]
```

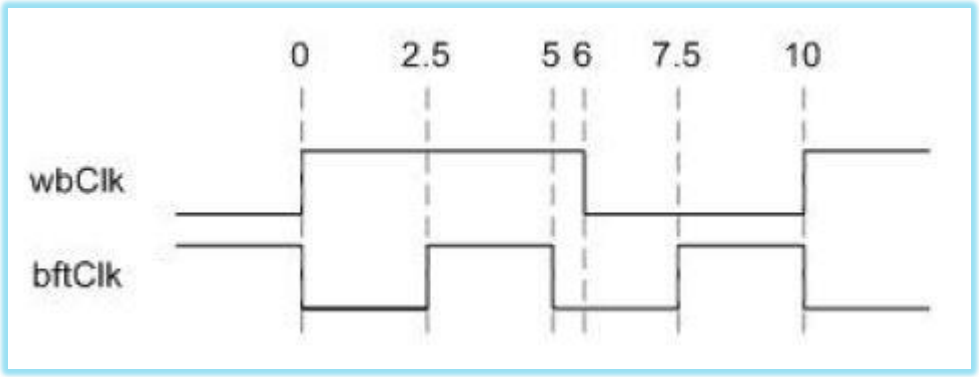
- The more specific the constraint, the higher the priority

```
set_max_delay -from [get_clocks clkA] -to [get_pins inst0/D] 12
set_max_delay -from [get_clocks clkA] -to [get_clocks clkB] 10
```

Win

Example

- **You have a design with two clocks coming on ports called wbClk and bftClk**
  - wbClk is a 100MHz clock, with 150 ps of jitter and a 60/40 duty cycle
  - within the wbClk clock domain, setup clock uncertainty 213 ps
  - bftClk is a 200MHz clock, with 30 ps of jitter and a 50/50 duty cycle
  - The falling edge of bftClk is aligned with the rising edge of wbClk
  - The design was designed to handle all CDC paths correctly. Assume that all CDC paths derived from the two primary clocks can be ignored



Example Solutions

Solutions

```
create_clock -name wbClk -period 10.0 -waveform {0.0 6.0} [get_ports wbClk]
set_input_jitter wbClk 0.15
set_clock_uncertainty -setup 0.213 [get_clocks wbClk]
create_clock -name bftClk -period 5.0 -waveform {2.5 5} [get_ports bftClk]
set_input_jitter bftClk 0.03
set_clock_groups -async -name my_async_clks -group wbClk -group bftClk
```

Example : Clock Validation

- Validate both clocks are defined successfully
  - `report_clocks`
- Validate input jitter is defined successfully
  - `report_property [get_clocks wbClk]`
  - `get_property INPUT_JITTER [get_clocks wbClk]`

```
Attributes
  P: Propagated
  G: Generated
  V: Virtual
  I: Inverted

Clock   Period    Waveform            Attributes  Sources
wbClk   10.00000  {0.00000 6.00000}   P           {wbClk}
bftClk  5.00000   {2.50000 5.00000}   P           {bftClk}
```

```
Property            Type     Read-only  Value
CLASS               string   true       clock
INPUT_JITTER        double   true       0.150
IS_GENERATED        bool     true       0
IS_PROPAGATED       bool     true       1
IS_USER_GENERATED   bool     true       0
IS_VIRTUAL          bool     true       0
NAME                string   true       wbClk
PERIOD              double   true       10.000
SOURCE_PINS         string*  true       wbClk
SYSTEM_JITTER       double   true       0.050
WAVEFORM            double*  true       0.000 6.000
```

Example : Clock Validation

- Validate clock uncertainty is defined successfully
  - `report_timing -from [get_clocks wbClk] -to [get_clocks wbClk]`

```
Slack (MET) :             7.116ns  (required time - arrival time)
  Source:                 egressLoop[0].egressFifo/buffer_fifo/infer_fifo.block_ram_performance.fifo_ram_reg/CLKBWRCLK
                            (rising edge-triggered cell RAMB36E1 clocked by wbClk  {rise@0.000ns fall@6.000ns period=10.000ns})
  Destination:            wbOutputData_reg[0]/D
                            (rising edge-triggered cell FDRE clocked by wbClk  {rise@0.000ns fall@6.000ns period=10.000ns})
  Path Group:             wbClk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            10.000ns  (wbClk rise@10.000ns - wbClk rise@0.000ns)
  Data Path Delay:        2.622ns  (logic 1.886ns (71.940%)  route 0.736ns (28.060%))
  Logic Levels:           2  (LUT6=2)
  Clock Path Skew:        -0.023ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    1.629ns = ( 11.629 - 10.000 )
    Source Clock Delay      (SCD):    1.760ns
    Clock Pessimism Removal (CPR):    0.108ns
  Clock Uncertainty:      0.296ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE + UU
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.150ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns
    User Uncertainty         (UU):    0.213ns
```

Example : Clock Validation

- Validate both clocks are asynchronous
  - `report_clock_interaction`

From Clock	To Clock	WNS Clock Edges	WNS	TNS	TNS Failing Endpoints	TNS Total Endpoints	WNS Path Requirement	Common Primary Clock	Inter-Clock Constraints
bftClk	bftClk	rise - rise	0.22	0.00	0	7705	5.00	Yes	Timed
bftClk	wbClk				0	33		No	Asynchronous Groups
wbClk	bftClk				0	440		No	Asynchronous Groups
wbClk	wbClk	rise - rise	7.33	0.00	0	1803	10.00	Yes	Timed